Electronic Version
Stylesheet Version v1.1.1

Claims

[c1] 1.In an integrated circuit having multiple processors with each processor having a ROM (read only memory) for storing operating code for controlling operation of the processor, a system for replacing defective operating code stored in the ROM with replacement code fixes comprising:

a RAM (random access memory) for storing replacement code fixes to replace the defective operating code;

a code fix local memory for each of the processors;

a code fix bus coupled to each of the code fix local memories and the RAM; and

means for loading the replacement code fixes over the code fix bus into all of the code fix local memories whenever one of the processors encounters a predetermined jump code during execution of the defective operating code.

[c2] 2. The system of claim 1, wherein each of the multiple processors utilizes the same operating code, and a first processor encountering a code fix requests the code fix from the RAM which distributes the code fix over the code fix bus to the code fix local memories, such that

each of the processors has access to the code fix from its own code fix local memory which stores the distributed code fix.

[c3] 3. The system of claim 1, wherein the RAM is loaded with replacement code fixes from an EEPROM (Electronically Erasable/Programmable ROM) at power-up.

[c4] 4. The system of claim 1, wherein the integrated circuit is fabricated on a chip and the RAM is on-chip.

[c5] 5. The system of claim 4, wherein the on-chip RAM is connected to an off-chip EEPROM (Electronically Erasable/Programmable ROM) that loads corrected code fixes to the on-chip RAM at power-up.

[c6] 6. The system of claim 1, wherein when a processor encounters a predetermined jump code during execution of the defective code, the processor checks its code fix local memory to determine if a replacement code fix has been loaded into its code fix local memory, and if not, the processor then makes a request to the jump address in the RAM, and in response to the request, the RAM outputs the replacement code fix on the code fix bus to all of the code fix local memories which store the replacement code fix.

[c7] 7. The system of claim 1, wherein the code fix bus is a

dedicated code fix bus that only transmits code fixes to the code fix local memories.

[c8] 8. The system of claim 7, including a plurality of dedicated code fix buses, and wherein the RAM includes four ports and has a dedicated code fix bus coupled to each of the four ports.

[c9] 9. The system of claim 1, wherein each code fix local memory comprises a code fix local cache.

[c10] 10. The system of claim 1, wherein each code fix local memory implements a least recently used code fix algorithm wherein when the code fix local memory is loaded with a corrected code fix from the code fix bus, the code fix local memory keeps track of the least recently used code fix, and when there is no room in the code fix local memory to store an additional code fix, the code fix local memory replaces the least recently used code fix with a new code fix received over the code fix bus.

[c11] 11. A method for replacing defective operating code, for controlling operation of a processor and stored in a ROM (read only memory), with replacement code fixes, in an integrated circuit having multiple processors with each processor having a ROM for storing operating code, the method comprising:

storing replacement code fixes in a RAM (random access memory) to replace the defective operating code;
storing replacement code fixes in a code fix local memory for each of the processors coupled by a code fix bus to the RAM; and
loading the replacement code fixes over the code fix bus into all of the code fix local memories whenever one of the processors encounters a predetermined jump code during execution of the defective operating code.

[c12] 12. The method of claim 11, wherein each of the multiple processors utilizes the same operating code, and a first processor encountering a code fix requests the code fix from the RAM which distributes the code fix over the code fix bus to the code fix local memories, such that each of the processors has access to the code fix from its own code fix local memory which stores the distributed code fix.

[c13] 13. The method of claim 11, including loading the RAM with replacement code fixes from an EEPROM (Electronically Erasable/Programmable ROM) at power-up.

[c14] 14. The method of claim 11, including fabricating the integrated circuit and the RAM on a chip.

[c15] 15. The method of claim 14, including connecting the on-chip RAM to an off-chip EEPROM (Electronically Erasable/Programmable ROM), and loading corrected code fixes to the on-chip RAM at power-up.

[c16] 16. The method of claim 11, wherein when a processor encounters a predetermined jump code during execution of the defective code, the processor checks its code fix local memory to determine if a replacement code fix has been loaded into its code fix local memory, and if not, the processor then makes a request to the jump address in the RAM, and in response to the request, the RAM outputs the replacement code fix on the code fix bus to all of the code fix local memories which store the replacement code fix.

[c17] 17. The method of claim 11, including using a dedicated code fix bus that only transmits code fixes to the code fix local memories.

[c18] 18. The method of claim 17, including using a plurality of dedicated code fix buses, and wherein the RAM includes four ports, and coupling a dedicated code fix bus to each of the four ports.

[c19] 19. The method of claim 11, wherein each code fix local memory comprises a code fix local cache.

[c20] 20. The method of claim 11, including each code fix local memory implementing a least recently used code fix algorithm wherein when the code fix local memory is loaded with a corrected code fix from the code fix bus, the code fix local memory keeps track of the least recently used code fix, and when there is no room in the code fix local memory to store an additional code fix, the code fix local memory replaces the least recently used code fix with a new code fix received over the code fix bus.